

Project Management Using Point Graphs

Abbas K. Zaidi

Mashhood Ishaque

Alexander H. Levis

Conference on Systems Engineering Research

March 2007

Finish-Start Barrier

In CPM/PERT temporal constraints are allowed only between the end of an activity and start of the following activity. Thus could not represent **nominally parallel** activities.





Constraints

Interval A, B, C, D, E

Length [A] = 5

Length [B] = 5

Length [C] = 5

Length [D] = 2

Length [E] = 10

{ A Meets B
C Meets D

{ C Precedes B
eE Precedes eD

Representation

?

We propose a graph-based representation called Point Graphs that overcomes the shortcomings of traditional approaches to project management.



- Point Graphs
- Representation of Constraints
- Operations on Point Graphs
- Scheduling Algorithms

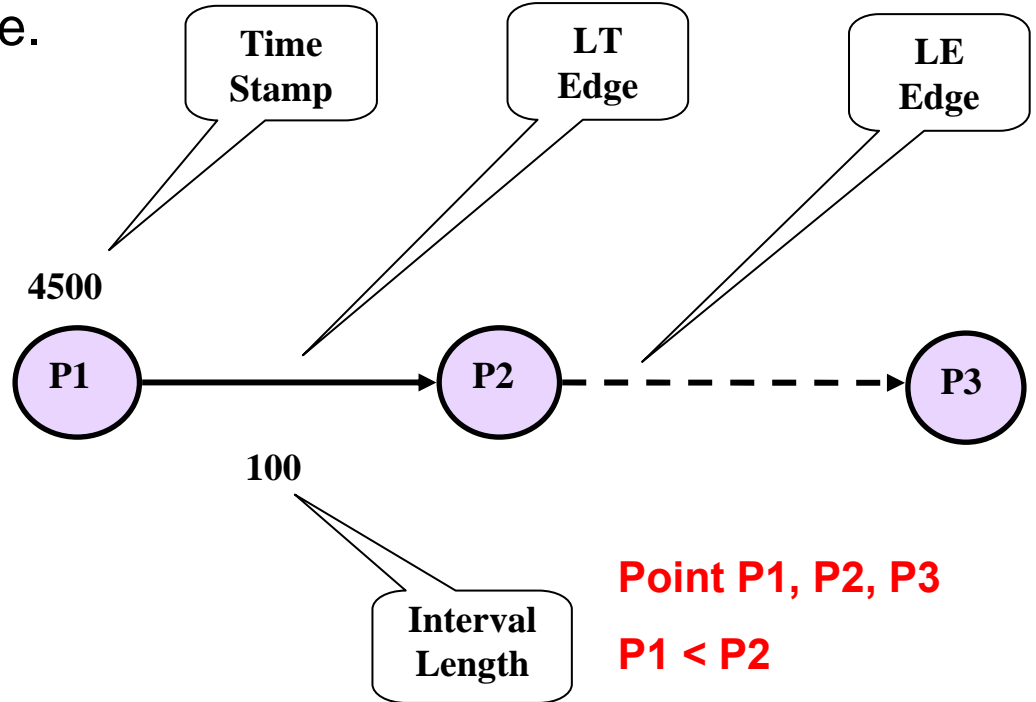
Node = point on the timeline.

Edge = temporal relation between points.

All LT Edges must have lengths specified for scheduling algorithms to work.

Knowledge Representation for Point-Interval Temporal Logic.

Inference using graph searches.



Point P1, P2, P3

$P1 < P2$



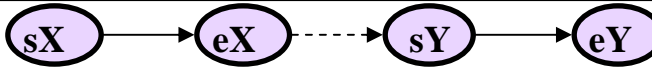

$P2 \leq P3$

Stamp [P1] = 4500



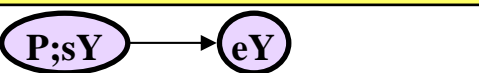



Length [P1,P2] = 100

Examples of Constraints (Qualitative)


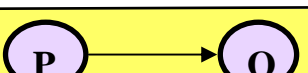




Interval X, Y	
X Before Y	
X Precedes Y	
X Meets Y	
...	

Interval to Interval Constraints

Point P, Interval Y	
P Before Y	
P Starts Y	
P During Y	
P Finishes Y	
Y Before P	


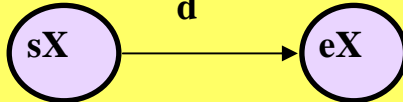
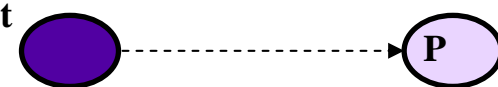
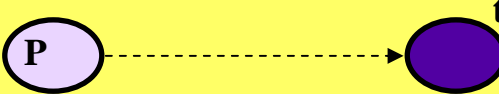
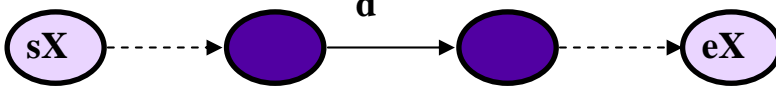
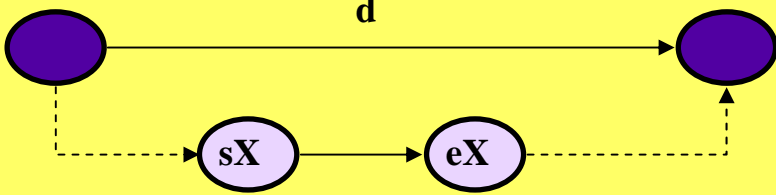
Point to Interval Constraints

Point P, Q	
P Before Q	
P Equals Q	
P Precedes Q	

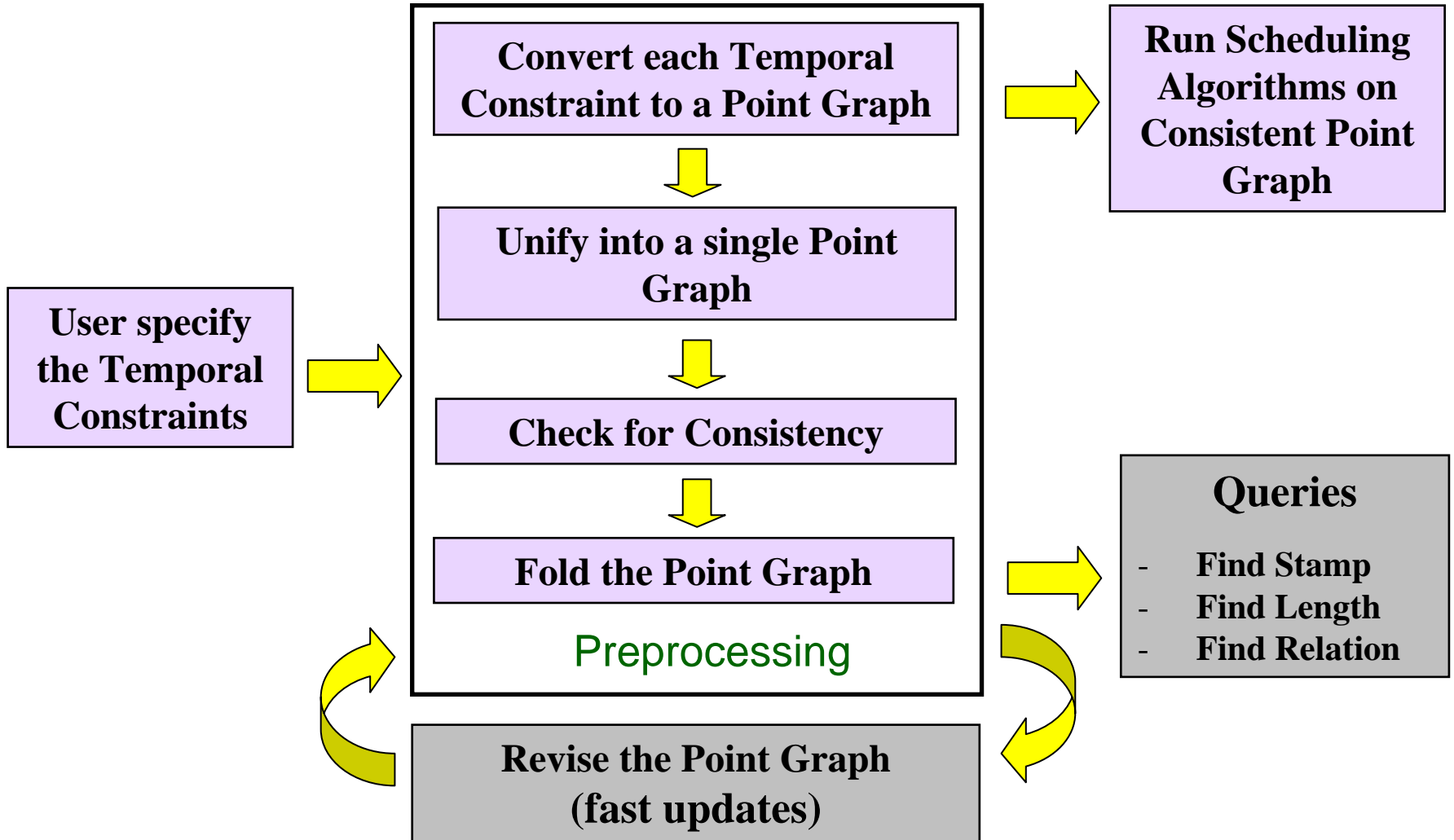
Point to Point Constraints

Examples of Constraints (Quantitative)



Stamp [P] = t	
Length [X] = d	
Stamp [P] ≥ t	
Stamp [P] ≤ t	
Length [X] ≥ d	
Length [X] ≤ d	

Stamp and Length Constraints

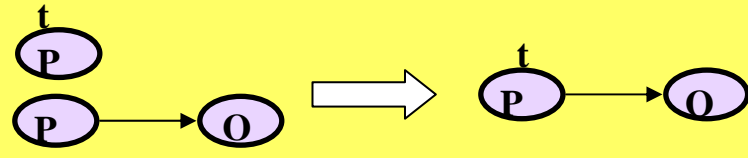


Operations on Point Graphs

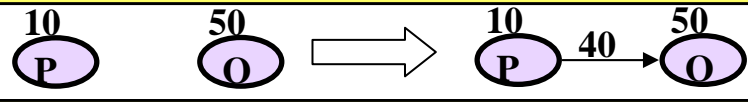


Unification

Unification I

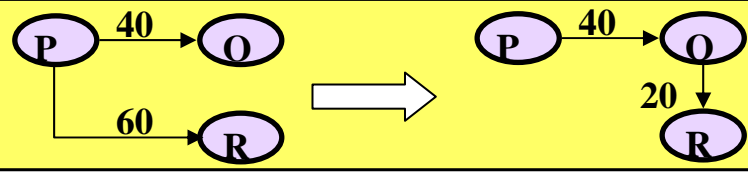


Unification II

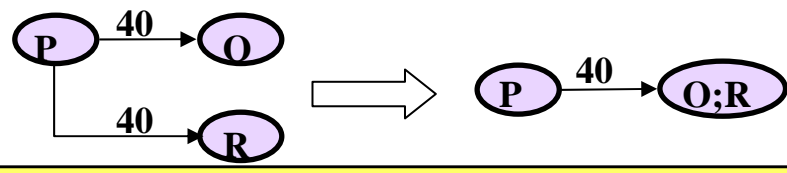


Folding

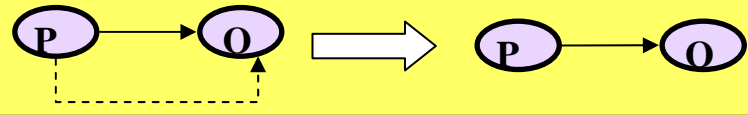
Branch Folding I



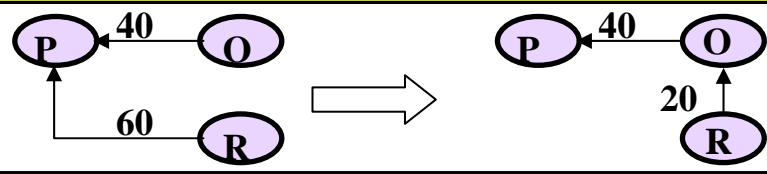
Branch Folding II



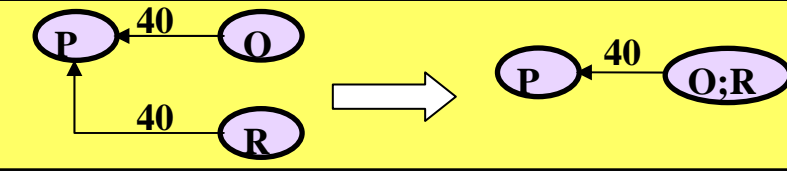
Branch Folding III



Join Folding I

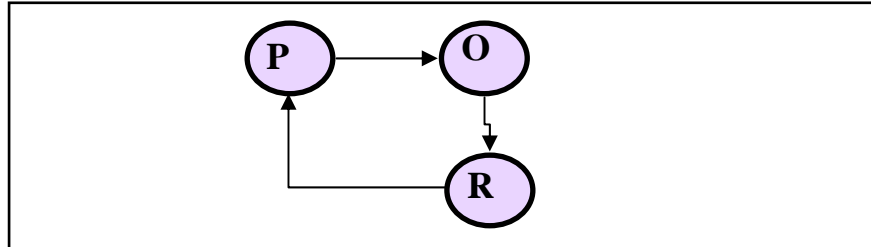


Join Folding II

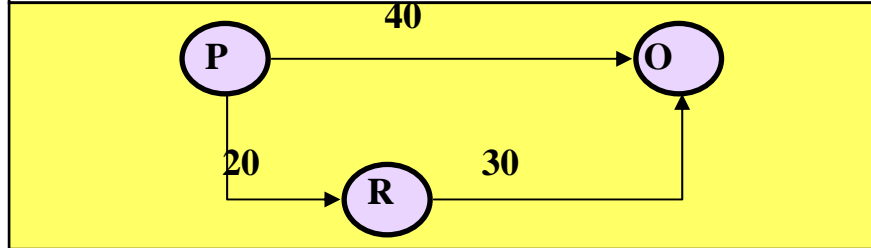


Verification

Check for Cycles

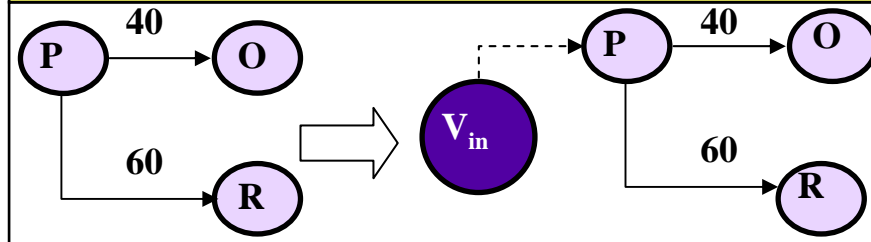


Check for Inconsistent Paths

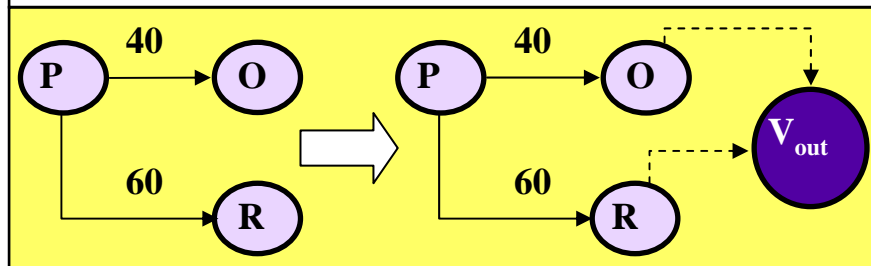


Virtual Nodes

Add Source



Add Sink



Constraints

Interval A, B, C, D, E

Length [A] = 5

Length [B] = 5

Length [C] = 5

Length [D] = 2

Length [E] = 10

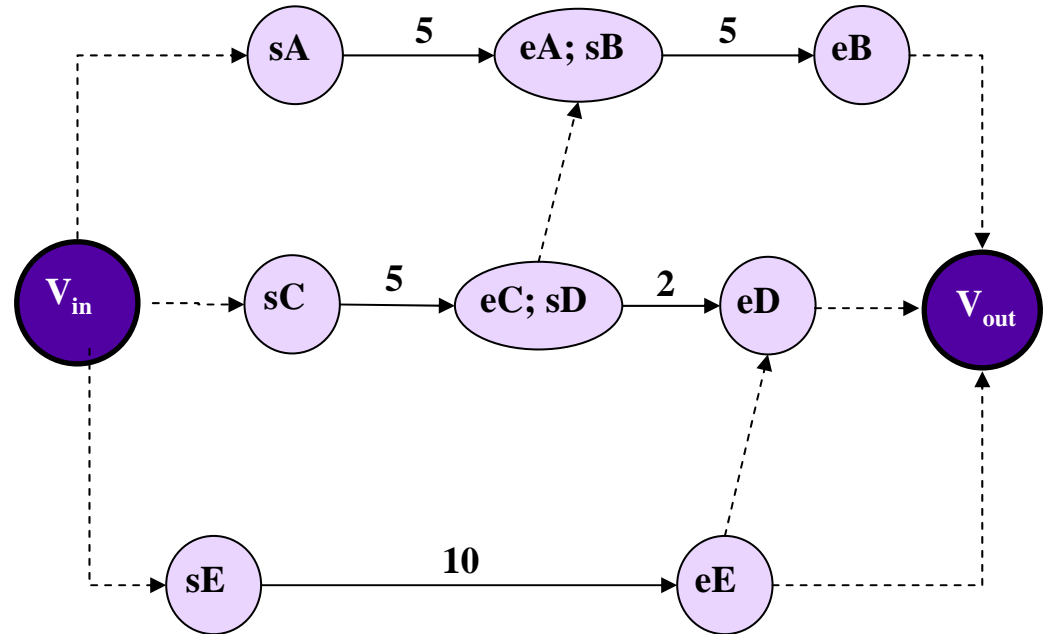
{ A Meets B

{ C Meets D

C Precedes B

{ eE Precedes eD

Point Graph



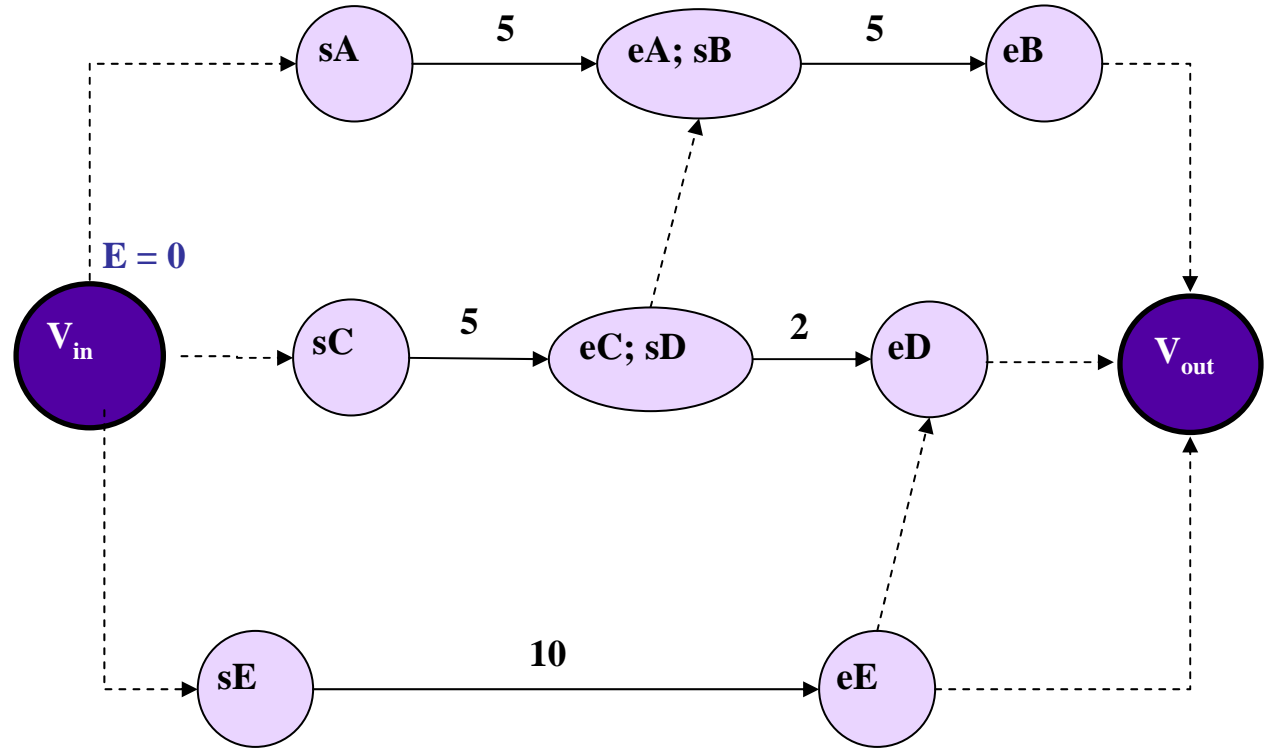


- Executed on Consistent Point Graph.
- Calculate Earliest Occurrence.
- Calculate Latest Occurrence.
- Identify Critical Activities.
- Identify Slacks/Floats for Non-Critical Activities.

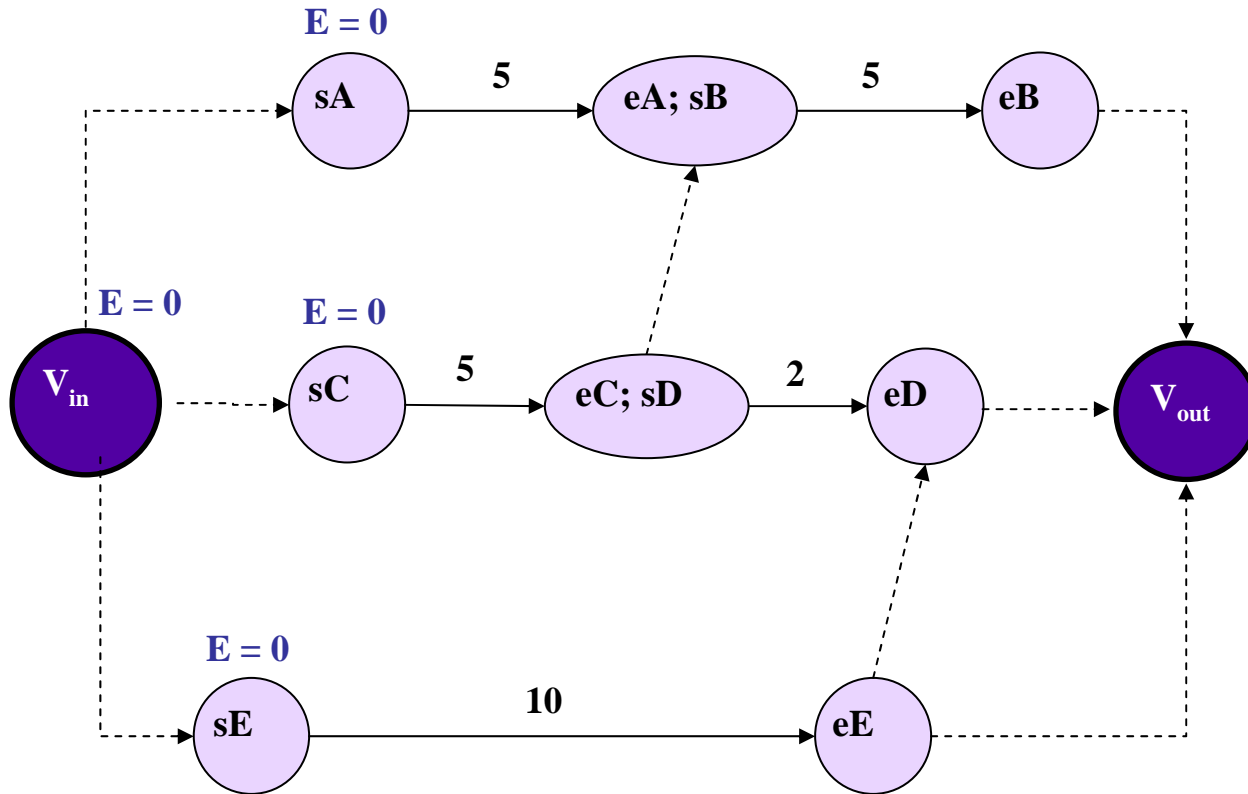


The earliest occurrence of a node is the smallest time stamp that can be assigned to it such that its preceding nodes can still start at their earliest.

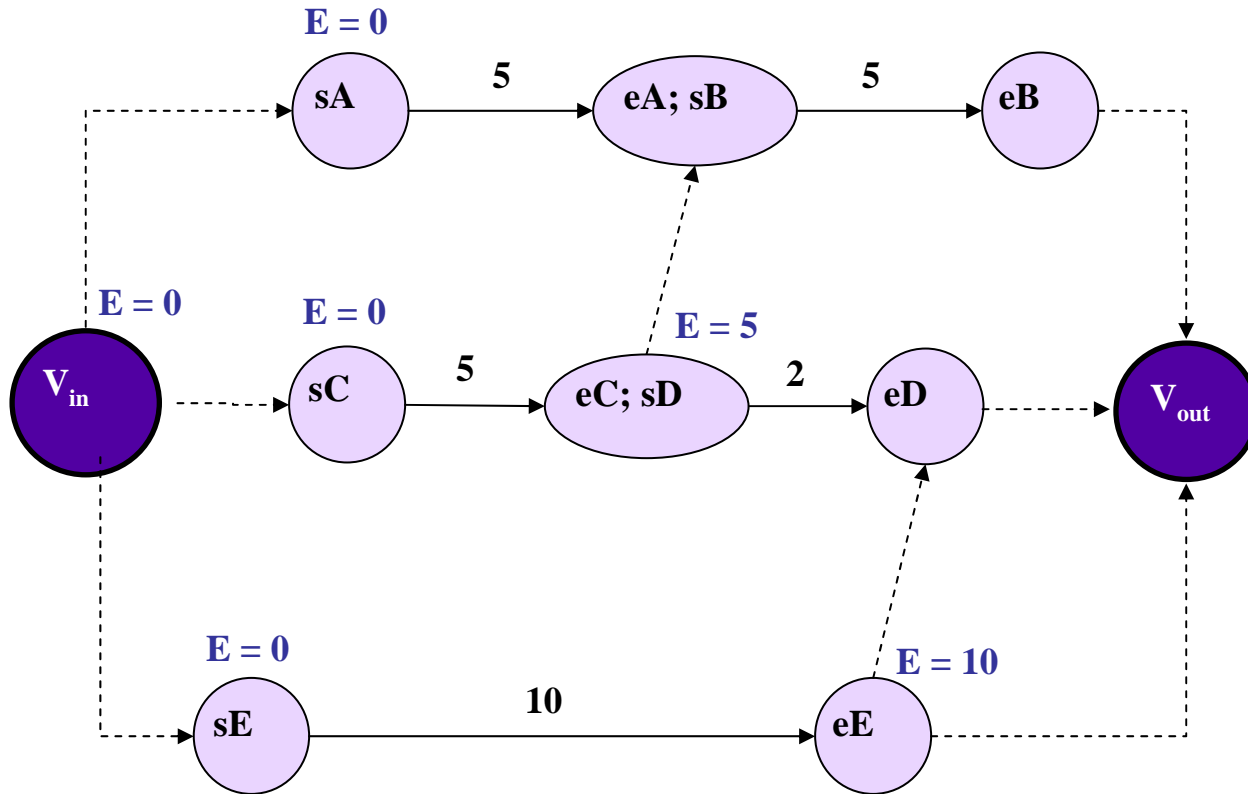
Earliest Occurrence



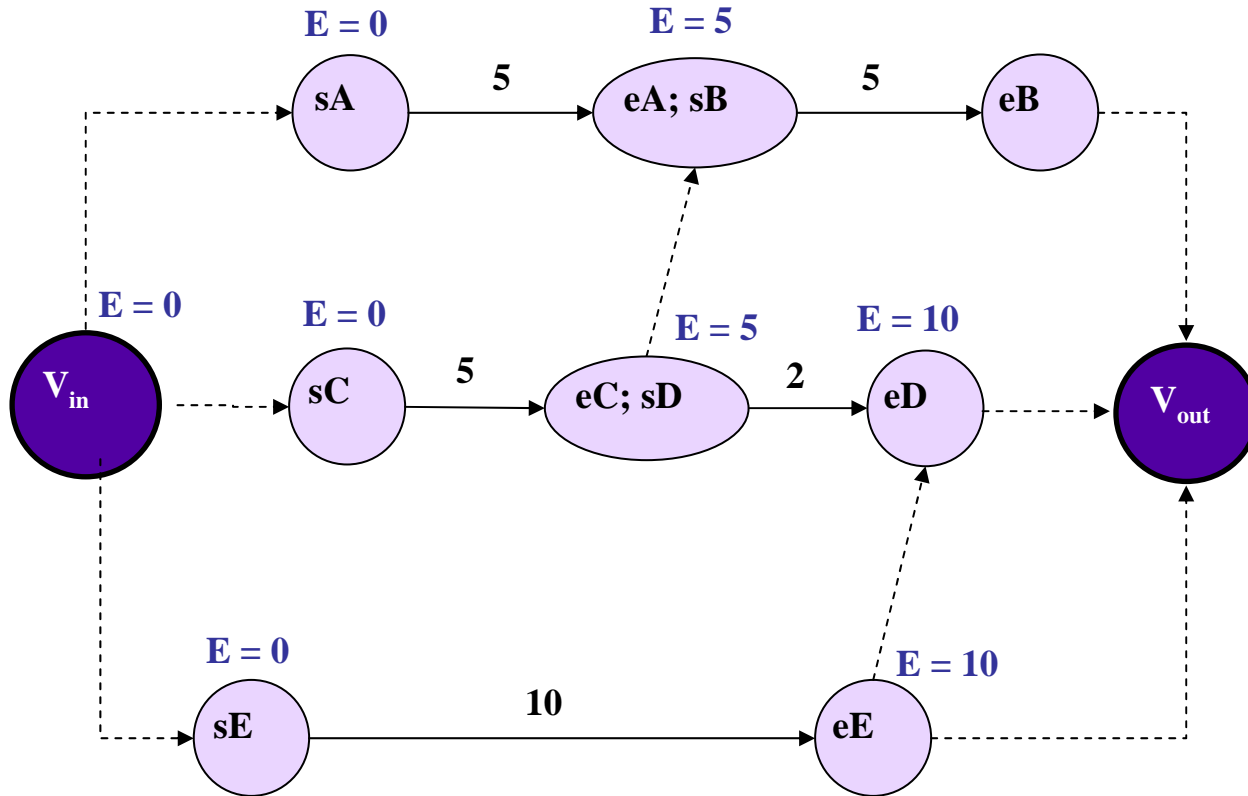
Earliest Occurrence



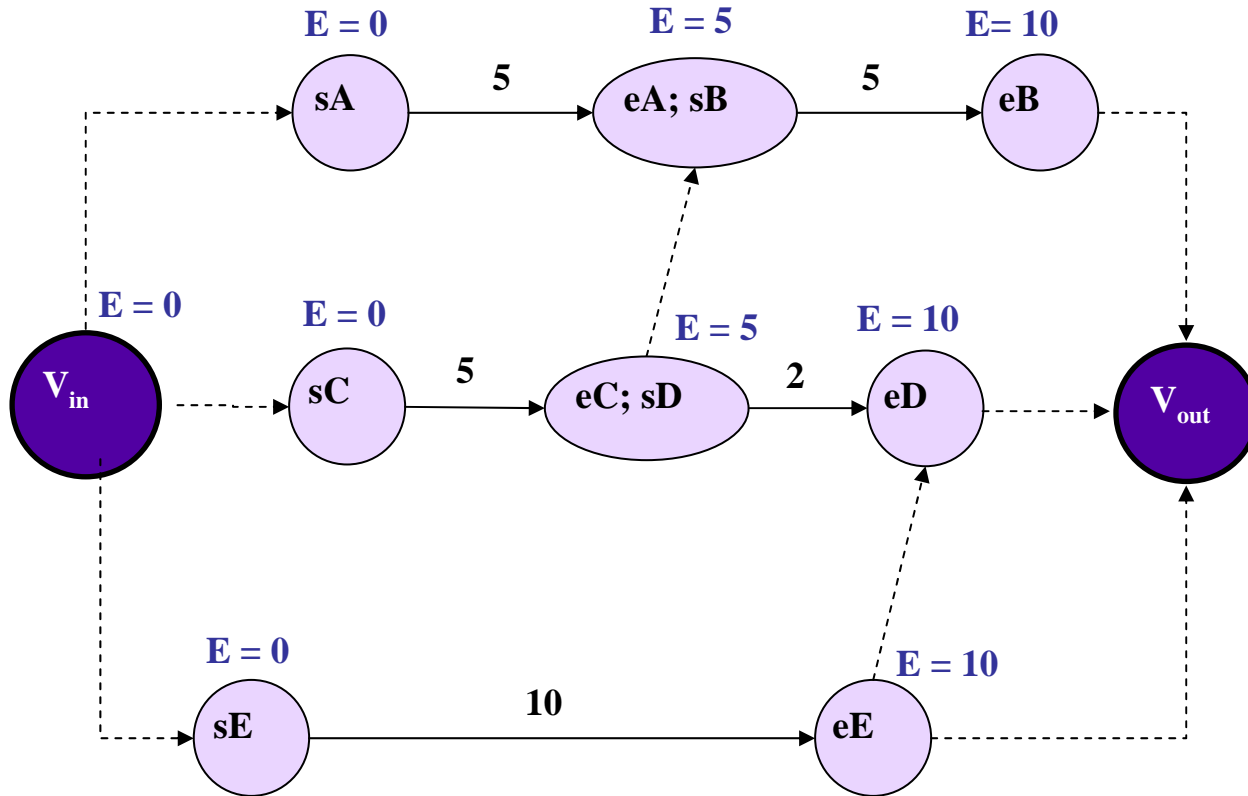
Earliest Occurrence



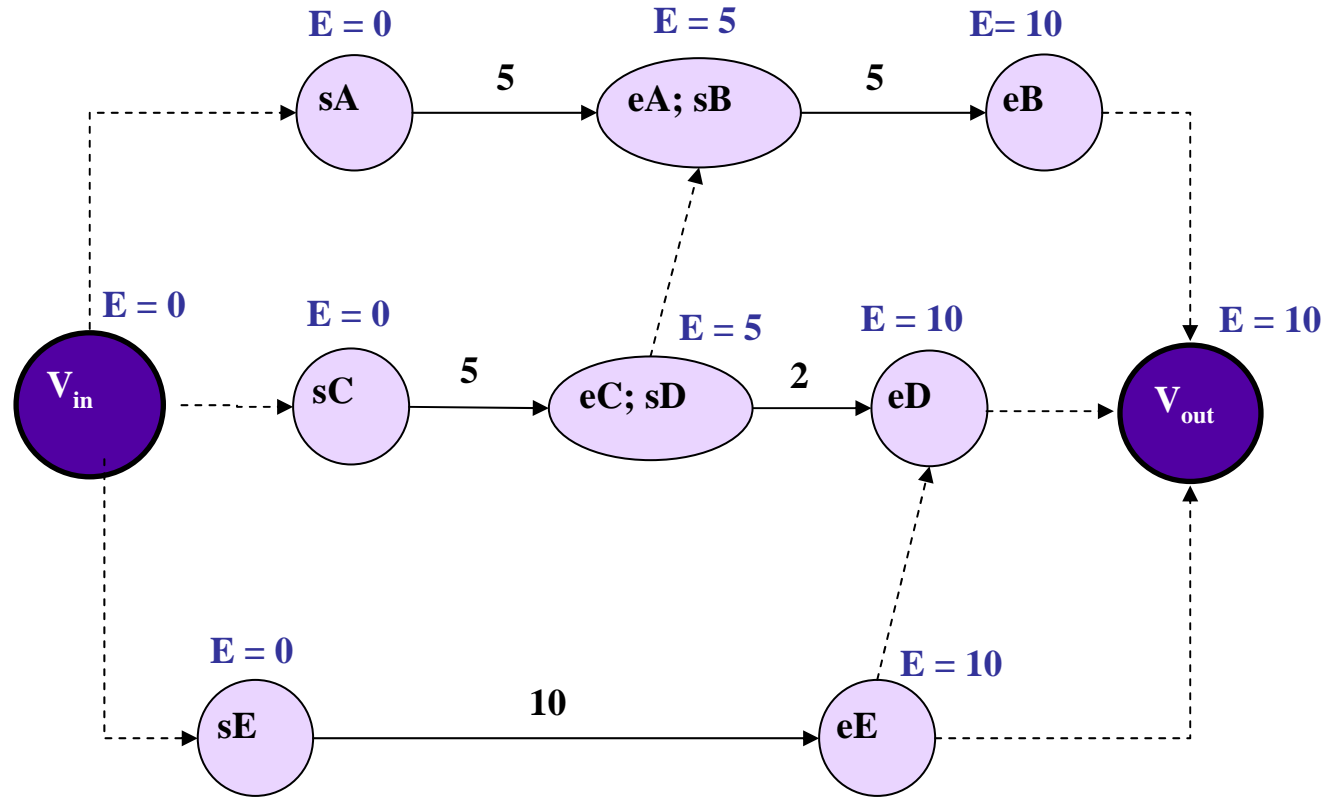
Earliest Occurrence



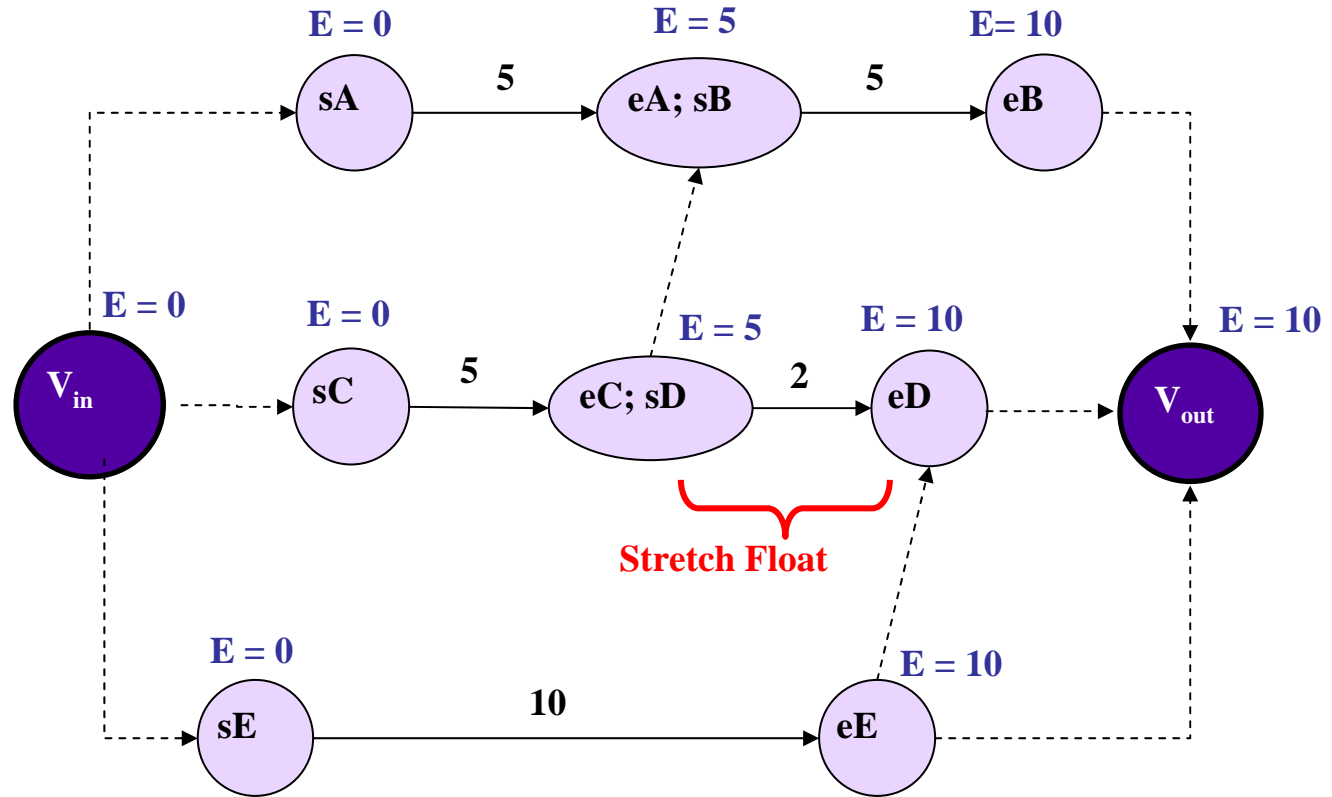
Earliest Occurrence



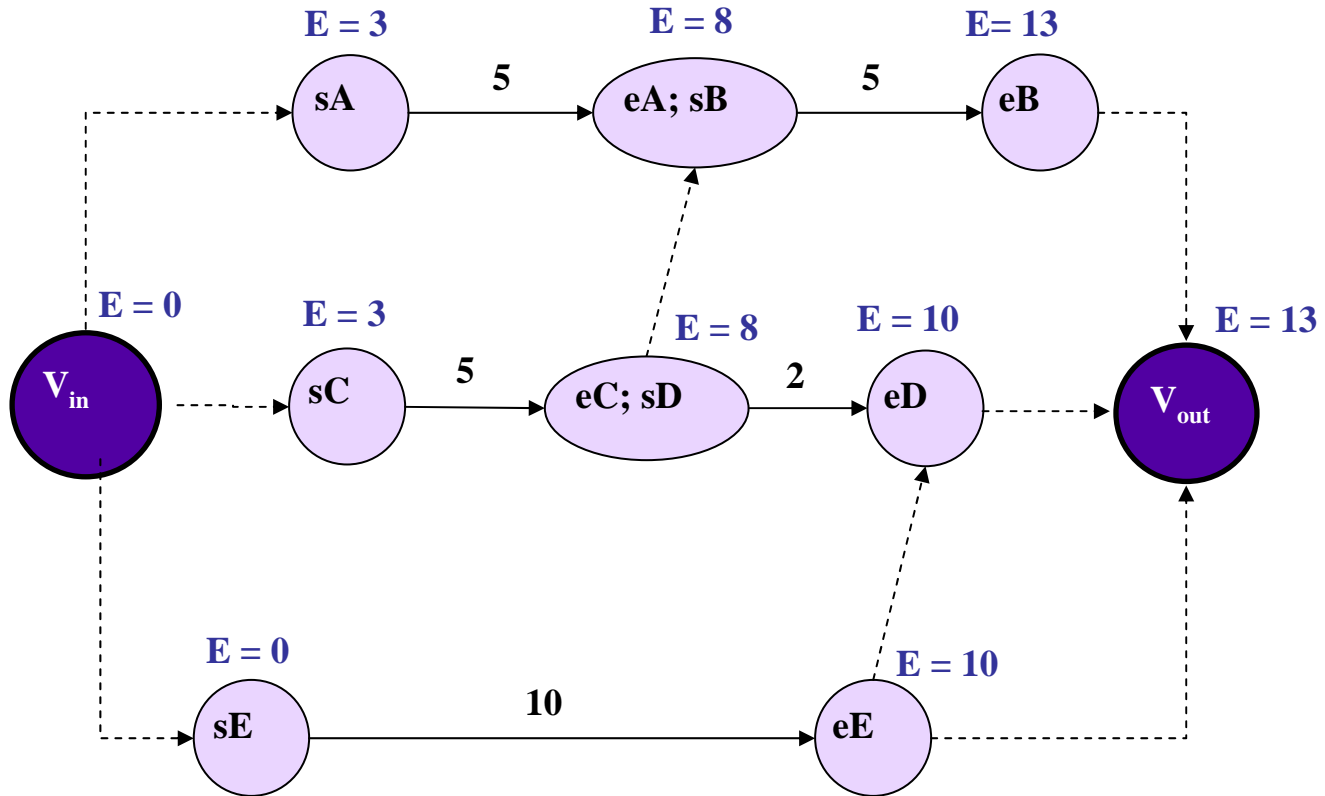
Earliest Occurrence



Earliest Occurrence

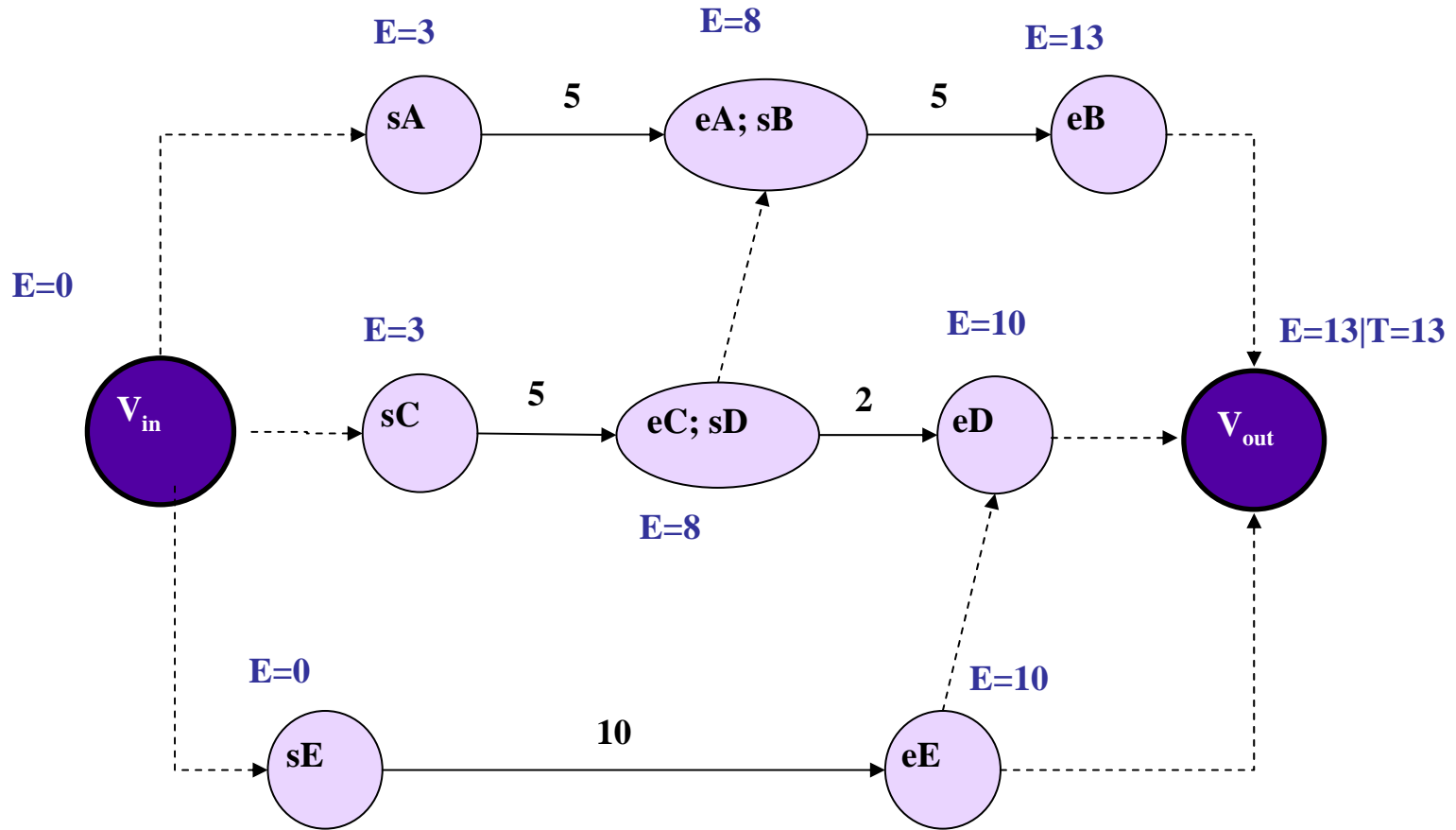


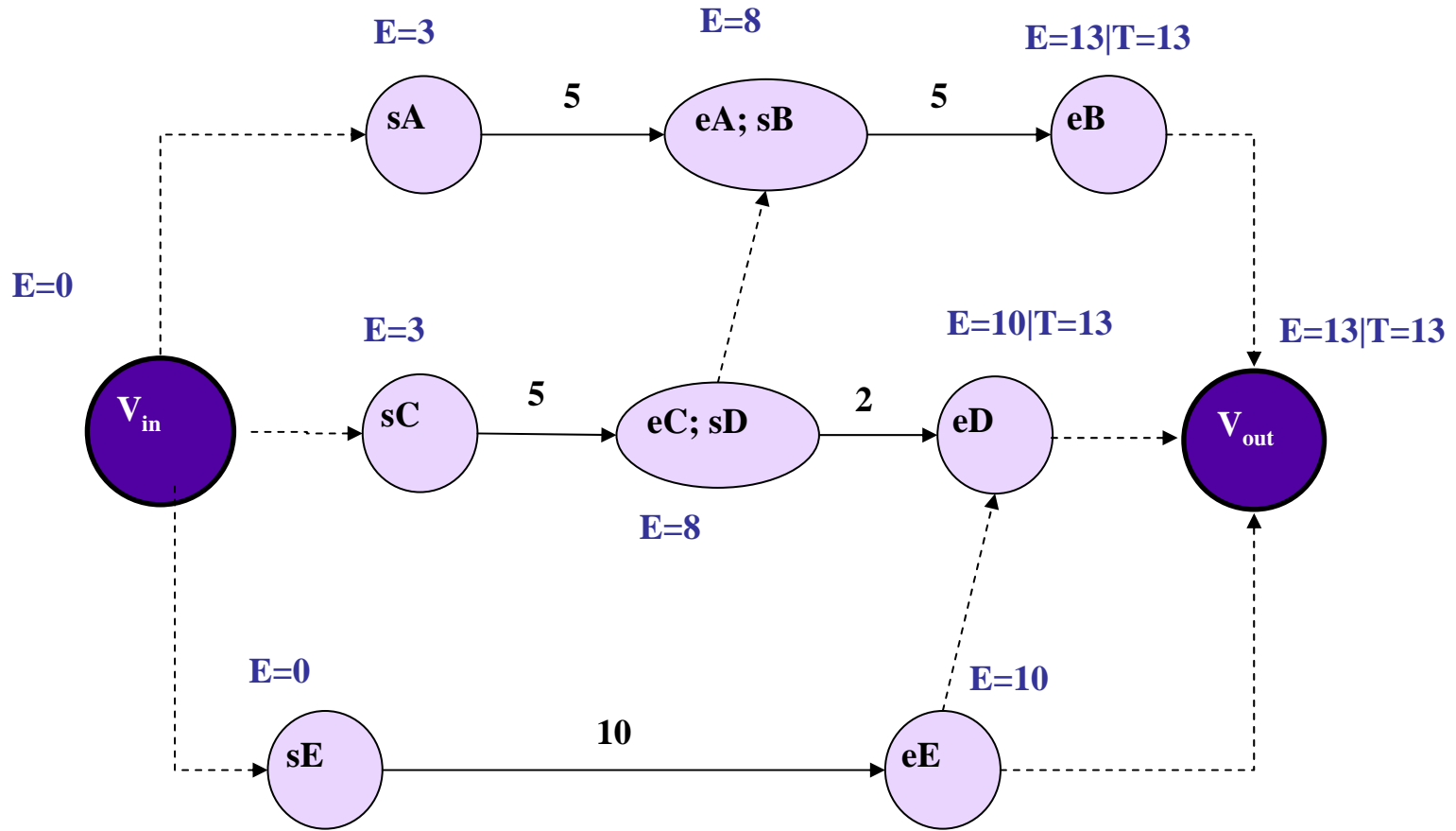
Earliest Occurrence

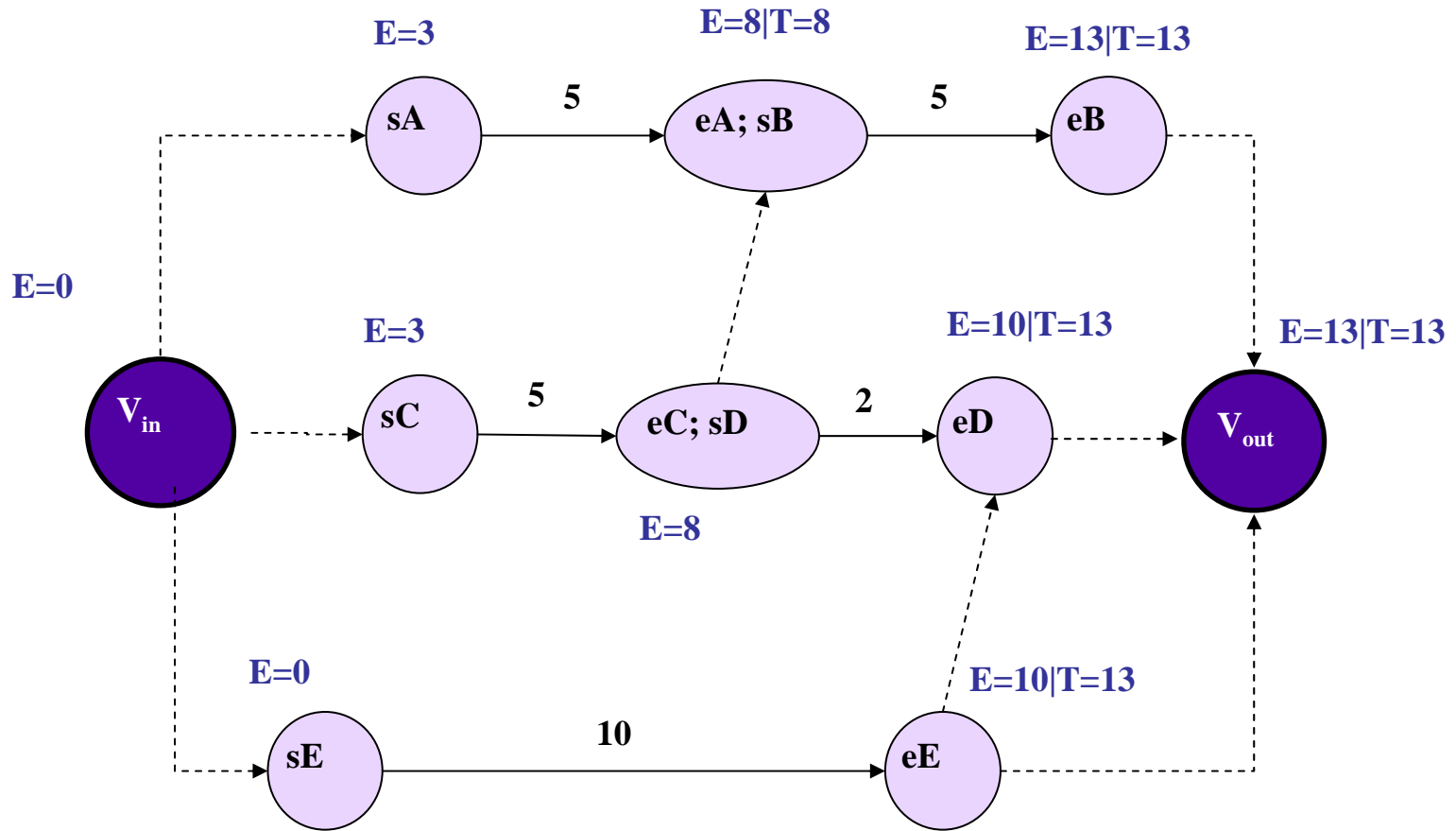


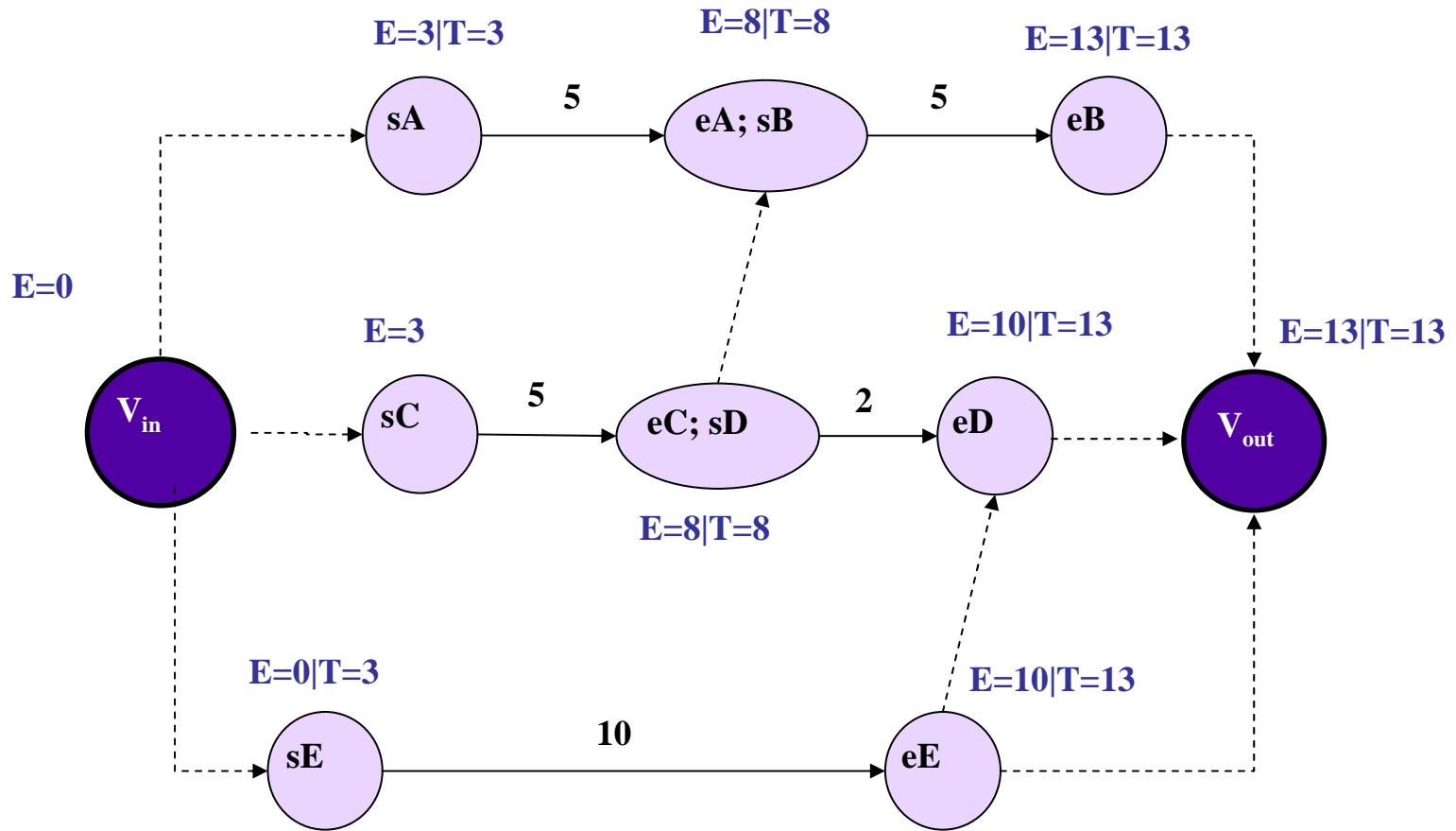


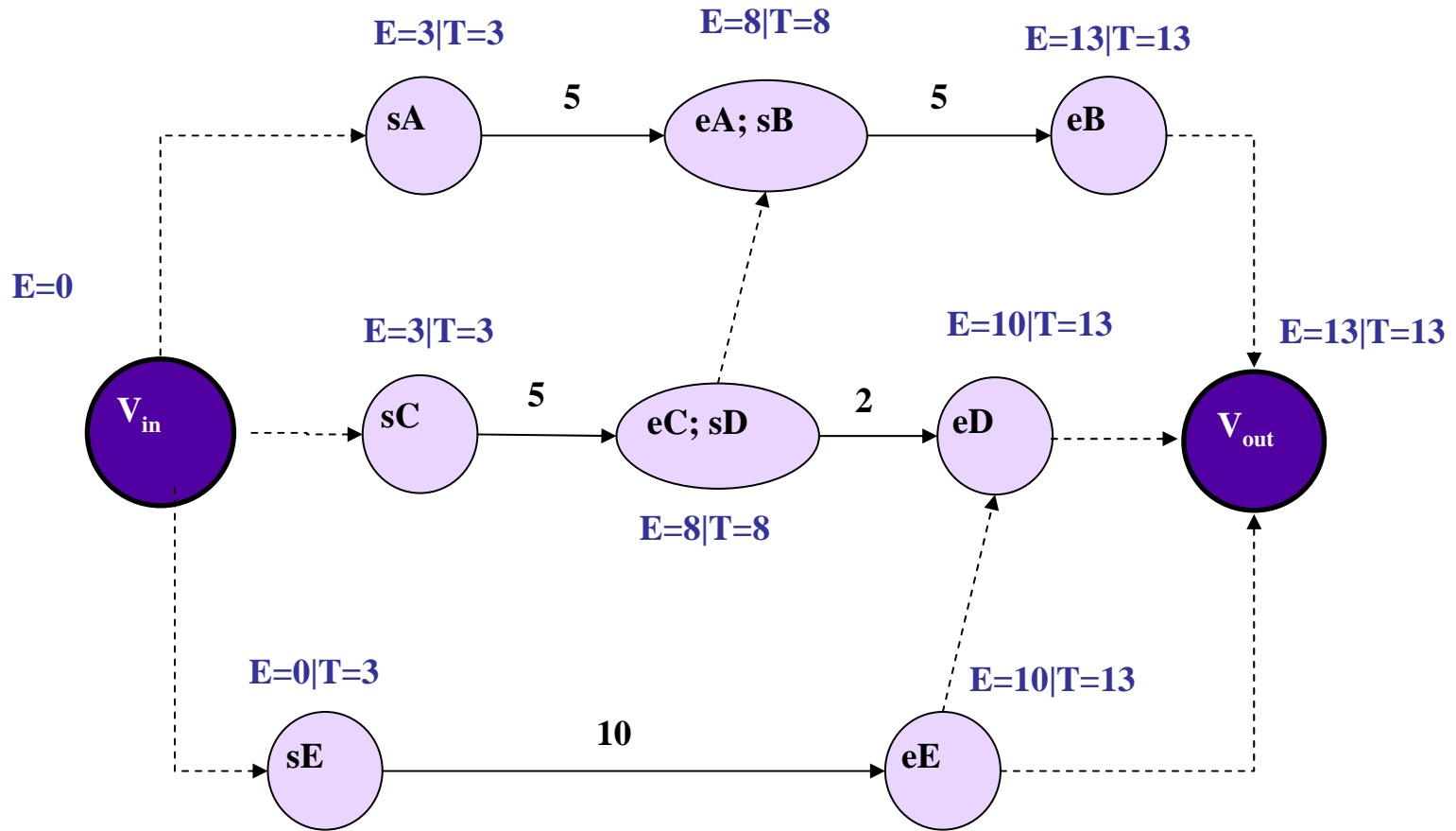
The latest occurrence of a node is the largest time stamp that can be assigned to it without increasing the completion time (minimum makespan) of the project.



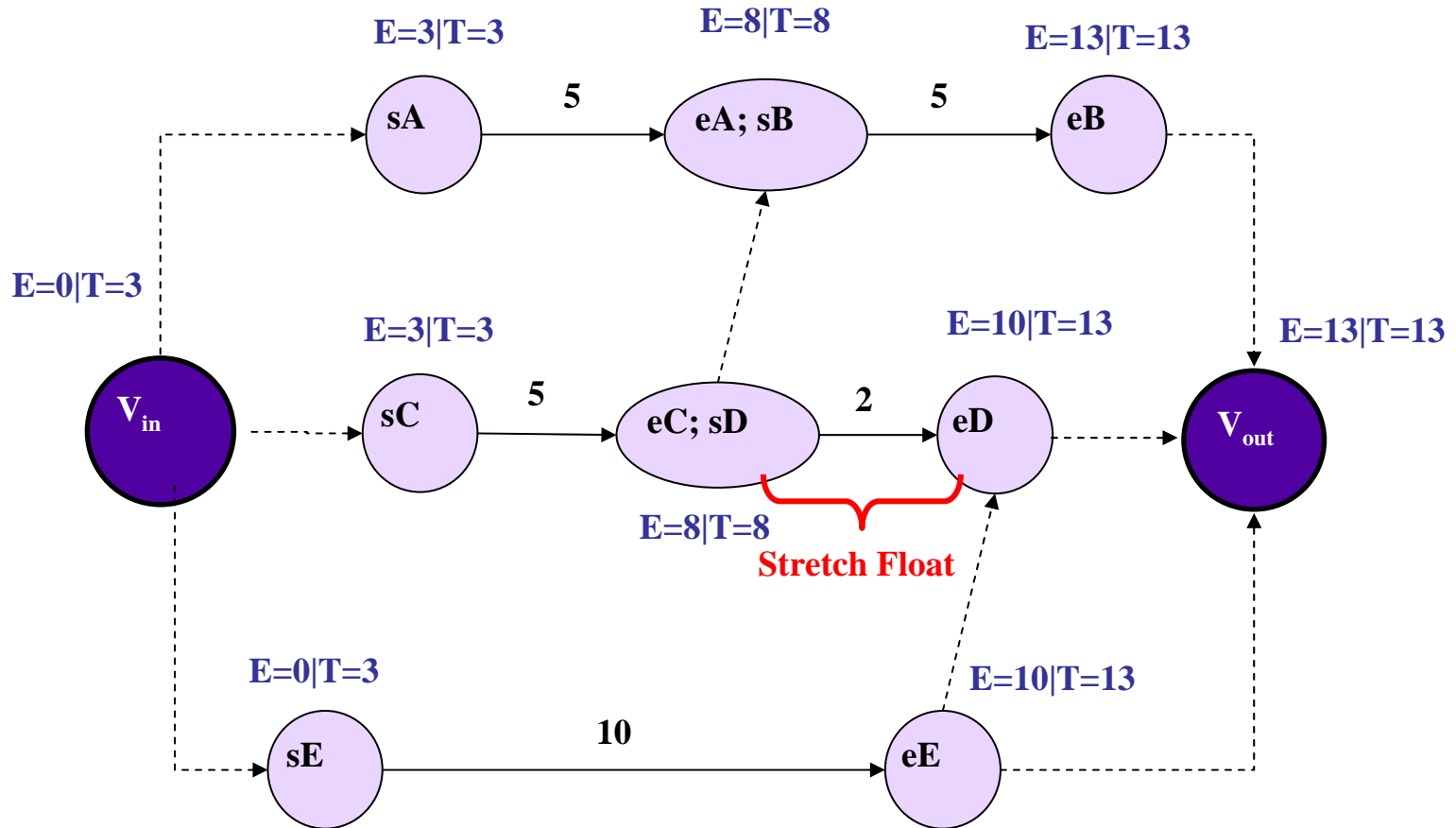


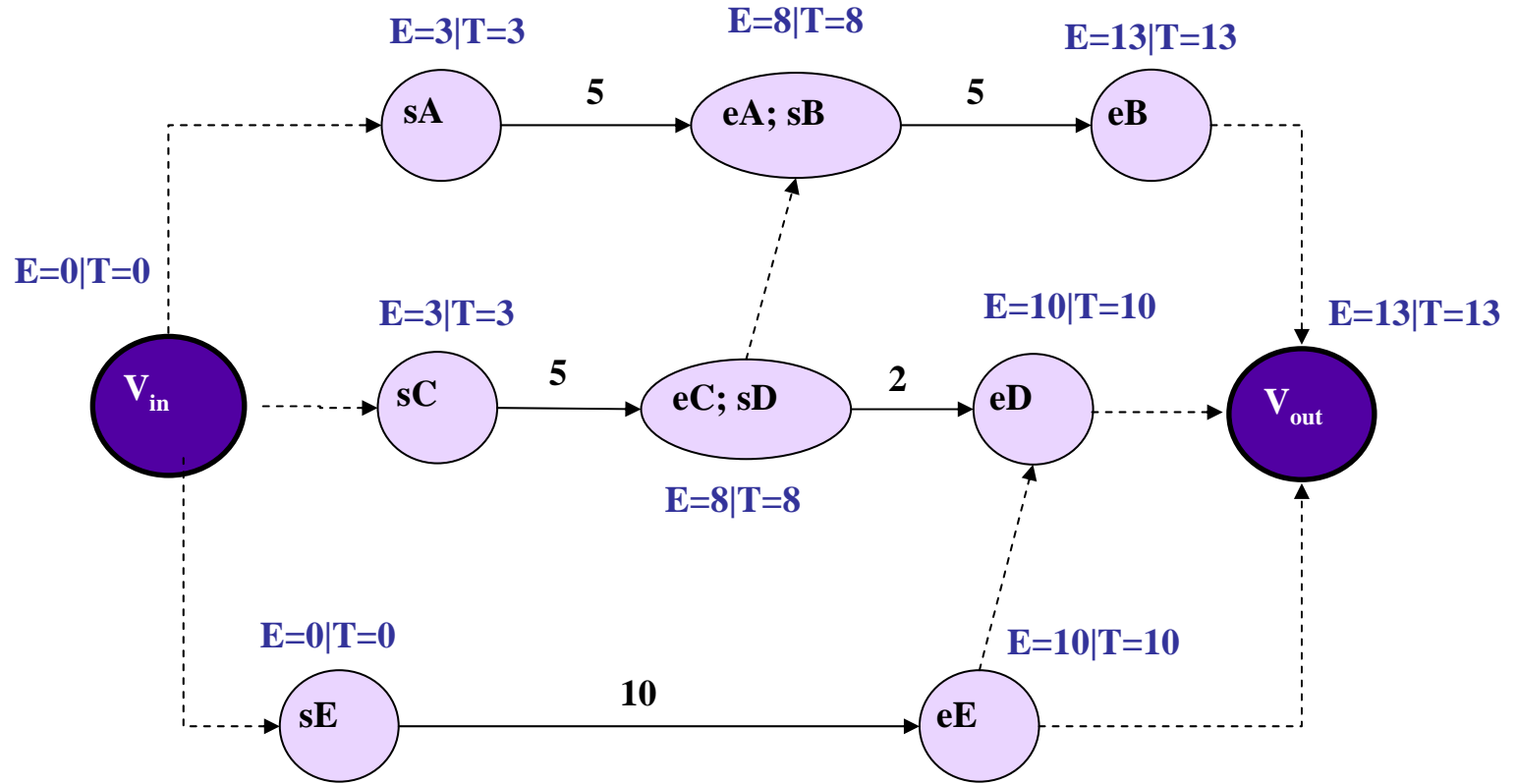




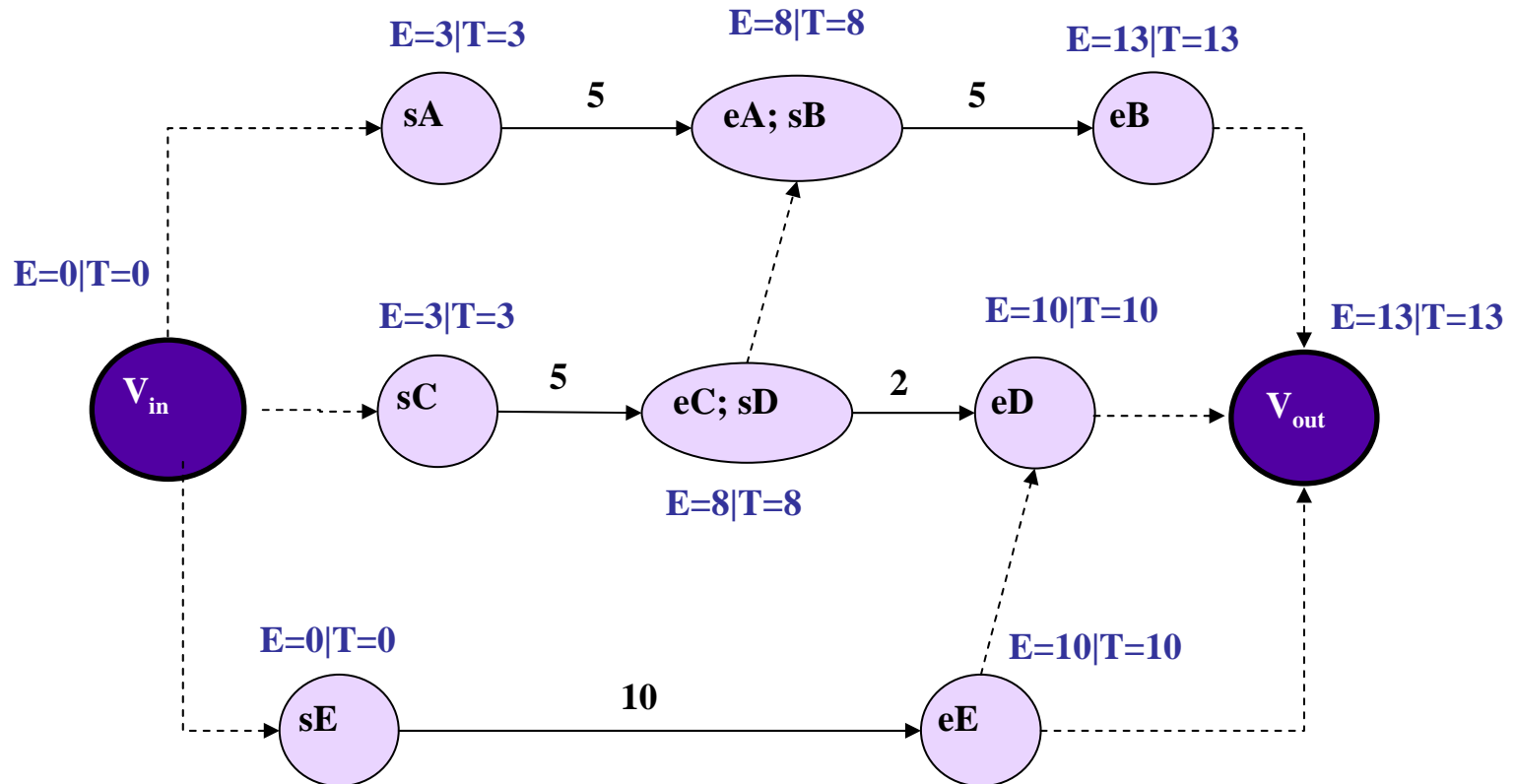


Latest Occurrence

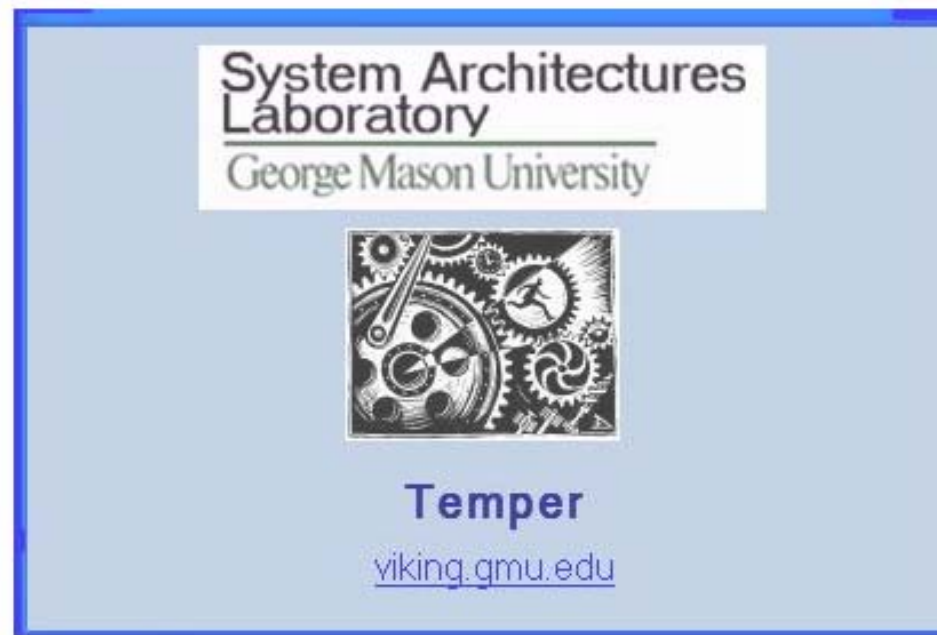




For critical activities the earliest occurrence is equal to latest occurrence.



The presented methodology has been implemented in a software called Temper.



Questions?



Thank You For Your Attention!